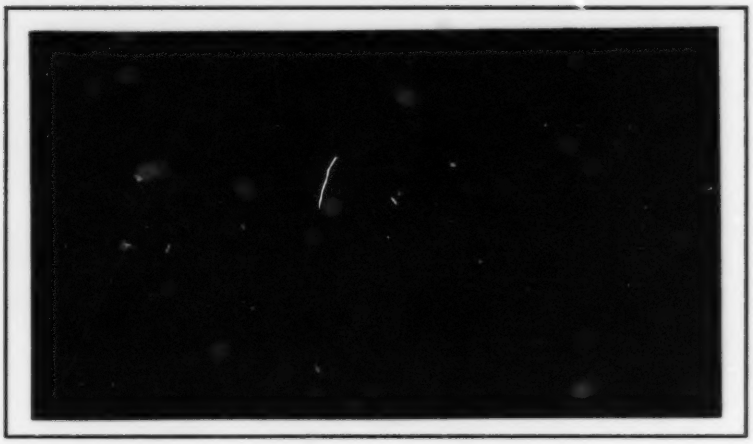

CENTRE FOR
SYSTEMS
SCIENCE

LCCR

Simon Fraser University, Burnaby, B.C., Canada V5A 1S6 / (604) 291-4704

Theory of Coteries

Toshihide Ibaraki and Tiko Kameda

CSS/LCCR TR90-09

August 1990
(Revised 1991)

92-05788/

Theory of Coteries: Mutual Exclusion in Distributed Systems*

Toshihide Ibaraki† and Tiko Kameda‡

Abstract

A coterie under a ground set U consists of a set of subsets (called quorums) of U such that any pair of quorums intersect each other. "Nondominated" coteries are of particular interest, since they are "optimal" in some sense. By assigning a boolean variable to each element in U, we represent a coterie by a boolean function of these variables. We characterize the nondominated coteries as exactly those which can be represented by positive, self-dual functions. In this boolean framework, we prove that any function representing an nondominated coterie can be composed from copies of the 3-majority function, and its composition is representable as a binary tree. We also show that the composition of nondominated coteries proposed by Agrawal and El Abbadi is a special case of the above tree composition, and that the composition proposed by Neilsen and Mizuno is closely related to the classical Ashenhurst decomposition of boolean functions. A number of other results are also obtained. The compactness of the proofs of most of these results indicates the suitability of boolean algebra to the analysis of coteries.

*This work was partially supported by grants from the Natural Sciences and Engineering Research Council of Canada, the Advanced Systems Institute of British Columbia, and by Grant-in-Aid from the Misistriy of Education, Science, and Culture of Japan.

†Department of Applied Mathematics and Physics, Kyoto University, Kyoto, JAPAN 606-01, Netaddress: ibaraki@kuamp.kyoto-u.ac.jp

‡School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6, Netaddress: tiko@cs.sfu.ca

1 Introduction and Definitions

1.1 History and Motivation

A coterie consists of subsets of an underlying set, such that any pair of subsets has at least one element in common. (For a more precise definition see Section 1.2.) The concept of coterie was introduced in [12] as a mathematical abstraction to model mutual exclusion in distributed systems [19, 27], in general, and a family of quorums in a replicated database system [13, 26], in particular.

Probably the best known example of a coterie consists of all the subsets consisting of $\lceil (n+1)/2 \rceil$ elements, where n is the total number of elements in the underlying set. This idea can be readily generalized to the "vote-assignable" coterie [12, 13, 27], in which each element u_i has weight w_i (positive integer) and a subset G belongs to the coterie if and only if it is a minimal subset satisfying $\sum_{u_i \in G} w_i > W/2$, where W is the sum of all the weights. The main motivation for generalizing this "majority" idea to general coteries was to have a large number of choices in implementing distributed exclusion, in order to select the most suitable one for the application in hand.

Consider an underlying set U, consisting of n elements. Any subset of these elements can be represented by an n-vector, $X = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$, where x_i is 1 (0) if the ith element is in the subset. In this paper, we represent a coterie by a boolean function $f(X)$ such that $f(X) = 1$ if and only if the argument X represents a superset of some subset (called a quorum) in the coterie. Using this representation, we can characterize an important class of coteries, called "nondominated (ND)" coteries [12]: a family of quorums forms an ND coterie if and only if $f(X)$ is positive and self-dual. Through this characterization, it becomes possible to utilize concepts and known results of boolean functions (see, e.g., [14, 21]).

Besides the class of vote-assignable ND coteries (see Section 6) discussed in [12], there are several known classes of ND coteries. Maekawa [19] showed that, based on finite projective planes, nontrivial ND coteries can be constructed such that each quorum contains only $\lceil \sqrt{n} \rceil$ elements. Agrawala and El Abbadi [1] proposed a new class of coteries based on complete binary trees. We generalize this class to "tree-decomposable" coteries, by removing the restrictions that the tree be complete and that each label be used only once. We then show that any ND coterie is tree-decomposable. This follows from one of our main theorems which states that any positive, self-dual function can be realized by a tree composed of copies of the basic majority function (i.e., the 3-input threshold function with the input weights equal to 1 and threshold equal to 2) one of whose inputs is a single variable. Another class of ND coteries has recently been proposed by Neilsen and Mizuno [22]. Viewed as boolean functions, this class turns out to correspond to a special case of the classical boolean function decomposition due to Ashenhurst [2], studied extensively in switching theory.

A vote-assignable coterie has the desirable property that it can be represented by a set of n weights. The only known general way of describing an arbitrary ND coterie, however, was to list all its quorums. This is not, in general, space-efficient, since a coterie can have an exponential number (in terms of the number of elements of the underlying set) of quorums. A composition of copies of the basic majority function that realizes a given positive, self-dual

function (e.g., the tree-decomposition mentioned above), on the other hand, often provides a compact method of describing an ND coterie.

Some concurrency control schemes for a replicated database system (e.g., [3, 7, 11, 15]) make use of a set of "read quorums" and a set of "write quorums" such that each read quorum intersects with each write quorum, but read quorums need not intersect each other. To model such a system, we introduce "bicoteries" and "semicoteries", and investigate their properties as well.

In the rest of this section we define the terms that are needed in this paper. Section 2 reviews, in the terminology of boolean algebra, a number of known results as well as those that follow easily from definitions. In Section 3, we investigate the relation among coteries, bicoteries, and semicoteries. We start the discussion of coterie decomposition in Section 4.1, and prove our main theorem mentioned above in Section 4.2. In Section 4.3, we consider the "non-redundantly decomposable" coteries, as a generalization of the class of coteries consisting of structured quorums [1]. Section 4.4 considers decomposition of coteries and its relevance to ND coteries. In Section 5, we consider another decomposition of ND coteries as a generalization of the method proposed in [22], and discuss its relevance to the Ashenhurst decomposition. Section 6 focuses on vote-assignable coteries and their generalization to multidimensional vote-assignable coteries introduced in [8].

In this paper, we sometimes omit the boolean AND operator $\wedge$, and simply concatenate variables to be ANDed together. $\subseteq$ and $\subset$ will denote set inclusion and proper set inclusion, respectively.

1.2 Definitions

Coterie[12]: Let $U = \{u_1, u_2, \ldots, u_n\}$ be a ground set and C be a family of subsets of U. C is a **coterie** under U if

(i) $G \in C \Rightarrow G \neq \phi$,

(ii) [**minimality**] There are no $G, H \in C$ such that $G \subset H$, and

(iii) [**intersection property**] $\forall G, H \in C$: $G \cap H \neq \phi$.

The subsets belonging to a coterie are called **quorums**. □

As a special case, a family C consisting of no subset is a coterie.

Domination[12]: Let C and D be coteries. C **dominates** D if $C \neq D$ and $\forall G \in D$ $\exists H \in C$: $H \subseteq G$. A coterie C is **nondominated** (ND) if no coterie dominates C. □

To see that ND coteries are particularly useful in database applications, let coterie D be dominated by C. Then if a partition (that results from a network partition failure) contains a quorum belonging to D, then it must contain a quorum belonging to C.

Bicoterie[3, 11]: An ordered pair $B = (P, Q)$, where P and Q are families of subsets of U, is a **bicoterie** under U if

(i) Both P and Q satisfy conditions (i) and (ii) for a coterie (with $C = P, Q$).

(ii) $\forall G \in P\ \forall H \in Q$: $G \cap H \neq \phi$.

A bicoterie $B = (P, Q)$ is called a **semicoterie** if it satisfies

(iii) $\forall G,\ H \in P$: $G \cap H \neq \phi$, (i.e., P is a coterie). □

A semicoterie $B = (P, Q)$ models the set of "read quorums" by Q and the set of "write quorums" by P. In [3], Q is called an anticoterie and $B = (P, Q)$ is called a quorum agreement.

Domination[3, 11]: Bicoterie $B = (P, Q)$ **dominates** bicoterie $B' = (R, S)$ if

(i) $\forall G \in R\ \exists H \in P$: $H \subseteq G$,

(ii) $\forall G \in S\ \exists H \in Q$: $H \subseteq G$, and

(iii) $(P, Q) \neq (R, S)$, i.e., $P \neq R$ or $Q \neq S$.

A bicoterie B is said to be **nondominated** (ND) if no bicoterie dominates B. Similarly, a semicoterie B is said to be **nondominated** (ND) if no semicoterie dominates B. □

In this paper, we assume that the boolean value 0 is less than the boolean value 1, i.e., $0 < 1$. For two boolean vectors $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$, we write $X \leq Y$ if $x_i \leq y_i$ for all $i = 1, 2, \ldots, n$.

Positive boolean function: A boolean function f is **positive** (or **monotone increasing**) if $\forall X, Y \in \{0,1\}^n$: $X \leq Y \Rightarrow f(X) \leq f(Y)$. □

Minimal vectors: Let f be a positive boolean function. $X \in \{0,1\}^n$ is a **minimal vector** of f if $f(X) = 1$ and $[Y \leq X \wedge Y \neq X] \Rightarrow f(Y) = 0$. The set of all minimal vectors (or the subsets they represent according to our convention) of function f will be denoted by $MIN(f)$. □

Let C be a family of distinct nonempty subsets of U satisfying the minimality condition. In this paper, we do not distinguish subset $G \in C$ from its characteristic vector, $(g_1, g_2, \ldots, g_n)$, where for $i = 1, 2, \ldots, n$,

$$g_i = \begin{cases} 1 & \text{if } u_i \in G \\ 0 & \text{if } u_i \notin G. \end{cases}$$

Define boolean function f_C associated with a family C of subsets of U by

$$f_C(X) = \begin{cases} 1 & \text{if } X \geq G \text{ for some } G \in C \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that $MIN(f_C) = C$ holds if C satisfies the minimality condition.

Property 1.1: The boolean function f_C defined above is positive for any family C. The correspondence between families C of subsets of U, satisfying the minimality condition, and positive boolean functions f_C is one-to-one. □

For two boolean functions f and g on the same set of variables, we write $f \leq g$ if and only if $\forall X \in \{0,1\}^n$: $f(X) = 1 \Rightarrow g(X) = 1$, and we write $f < g$ if and only if $f \leq g$ and $f \neq g$. The follwoing property is immediate from the definition of domination.

Property 1.2: Let C_1 and C_2 be two coteries under the same underlying set. Then C_1 dominates C_2 if and only if $f_{C_2} < f_{C_1}$. □

Remark 1.1: In discussing $f_C(X)$, we introduce a boolean variable x_i corresponding to each element $u_i \in U$. It is known (e.g., [21]) that any positive function f can be represented by a unique disjunctive form without complemented literals such that each term is a prime implicant corresponding to a minimal vector. This form is called the **minimum sum** of f. For f_C, it has the following form.

$$f_C = \bigvee_{G \in C} \{ \bigwedge_{u_i \in G} x_i \}.$$

□

Dual boolean function: For any boolean function f, its **dual**, denoted f^d, is defined by

$$f^d(X) = \bar{f}(\bar{X}),$$

where $\bar{X}$ is the complement[1] of $X \in \{0,1\}^n$, and $\bar{f}$ denotes the complement of f. □

Remark 1.2: Given an expression for function f, an expression for f^d is obtained by interchanging $\wedge$ and $\vee$. If f involves a constant 1 (0), it should be changed to 0 (1). For example, a positive function

$$f_C = \bigvee_{G \in C} \{ \bigwedge_{u_i \in G} x_i \}$$

has its dual

$$f_C^d = \bigwedge_{G \in C} \{ \bigvee_{u_i \in G} x_i \}.$$

(This relationship between f_C and f_C^d implies that $MIN(f_C^d)$ represents the set of all the minimal transversals [12] of C.) □

Property 1.3: Let f_C be the boolean function defined for a family C of nonempty subsets of U, satisfying the minimality condition. Then the dual of f_C is given by

$$f_C^d(X) = \begin{cases} 0 & \text{if } \bar{X} \geq G \text{ for some } G \in C \\ 1 & \text{otherwise.} \end{cases}$$

□

Property 1.4: If f is positive, then so is f^d. □

Self-duality: A boolean function f is said to be **dual-minor** if $f \leq f^d$, and **dual-major** if $f^d \leq f$. Note that a simple product term, e.g., $f = x_1x_2x_3$ is dual-minor, and its dual, $x_1 \vee x_2 \vee x_3$ is dual-major. f is said to be **dual-comparable** if $f \leq f^d$ or $f^d \leq f$ holds, and **self-dual** if $f = f^d$. □

Property 1.5:

(i) f is dual-minor $\Leftrightarrow \forall X \in \{0,1\}^n$: at least one of $f(X)$ and $f(\bar{X})$ takes value 0.

[1] The ith bit of vector $\bar{X}$ is 1 (0) if and only if the ith bit of X is 0 (1).

(ii) f is dual-major $\Leftrightarrow \forall X \in \{0,1\}^n$: at least one of $f(X)$ and $f(\bar{X})$ takes value 1.

(iii) f is self-dual $\Leftrightarrow \forall X \in \{0,1\}^n$: exactly one of $f(X)$ and $f(\bar{X})$ takes value 1. □

2 Basic Results

This section lists a number of basic results which either follow easily from definitions or are already known [11, 12]. The theorems below are based on the following properties of boolean functions. Let C be a family of nonempty subsets of an underlying set U.

(i) For any two subsets G and H: $G \cap H = \phi \Leftrightarrow G \leq \bar{H}$ $(= U - H)$ (viewed as boolean vectors). Therefore, for a family of subsets C satisfying the minimality condition,

$$\begin{aligned} &\forall G,\ H \in C\text{: } G \cap H \neq \phi \text{ (intersection property)} \\ &\qquad \Leftrightarrow \forall G,\ H \in C : G \not\leq \bar{H} \text{ and } H \not\leq \bar{G} \\ &\qquad \Leftrightarrow f_C \text{ is dual-minor.} \end{aligned}$$

The last line above follows from the second last line, since, by Property 1.5(i), if $f_C(X) = 1$ and $f_C(\bar{X}) = 1$ hold, i.e., $X \geq G$ for some $G \in C$ and $\bar{X} \geq H$ for some $H \in C$, then $\bar{G} \geq \bar{X} \geq H$, a contradiction. This proves $\Rightarrow$. The converse is also similar.

(ii) Let C_1 and C_2 be coteries. By Property 1.2, C_1 dominates C_2 if and only if $f_{C_2} < f_{C_1}$, or equivalently, $f^d_{C_1} < f^d_{C_2}$. Since f_{C_1} is dual-minor (by point (ii) above), we have $f_{C_1} < f^d_{C_1}$, hence $f_{C_2} < f^d_{C_2}$, i.e., f_{C_2} is not self-dual. Therefore

$$f_C \text{ is self-dual} \Rightarrow C \text{ is an ND coterie.}$$

Conversely, if f_C is dual-minor but not self-dual, then $f_C < f^d_C$ and there is a $Y \in \{0,1\}^n$ such that $f_C(Y) = 0$ and $f^d_C(Y) = 1$. Let $C' = C \cup \{Y\}$, where Y is regarded as a subset. Then $f_{C'}$ is dual-minor, since it satisfies Property 1.5(i) (in particular, $f_{C'}(\bar{Z}) = 0$ for any $Z \geq Y$, since $f^d_C(Y) = 1 \Rightarrow f_C(\bar{Y}) = 0 \Rightarrow f_C(\bar{Z}) = 0$ for any $\bar{Z} \leq \bar{Y} \Rightarrow f_{C'}(\bar{Z}) = 0$ for any $\bar{Z} \leq \bar{Y}$.) Furthermore, $f_C < f_{C'}$ holds, implying that C is not ND. Therefore

$$C \text{ is an ND coterie} \Rightarrow f_C \text{ is self-dual.}$$

Theorem 2.1: Let C be a family of nonempty subsets of U satisfying the minimality condition. Then C is a coterie if and only if f_C is dual-minor. □

Theorem 2.2: Let C be a family of nonempty subsets of U satisfying the minimality condition. Then C is an ND coterie if and only if f_C is self-dual. □

Corollary 2.1: Let C be a coterie under U. Then C is ND if and only if for $\forall X \subset U$ exactly one of X and $\bar{X}$ contains some subset $G \in C$. □

These properties can be proved easily for bicoteries and semicoteries as well.

Theorem 2.3: Let $B = (P, Q)$, where P and Q are families of nonempty subsets of U satisfying the minimality condition. Then B is a bicoterie if and only if $f_Q \leq f_P^d$ (or equivalently $f_P \leq f_Q^d$). Similarly, B is a semicoterie if and only if $f_Q \leq f_P^d$ and f_P is dual-minor. □

Theorem 2.4: Let $B = (P, Q)$ be as defined in Theorem 2.3. Then B is an ND bicoterie if and only if $f_Q = f_P^d$ (or equivalently $f_P = f_Q^d$). Also, B is an ND semicoterie if and only if $f_Q = f_P^d$ and f_P is dual-minor. □

We can summarize the main results in terms of $MIN(f)$.

(i) f is positive and self-dual $\Leftrightarrow$ $MIN(f)$ is an ND coterie.

(ii) f is positive $\Leftrightarrow$ ($MIN(f)$, $MIN(f^d)$) is an ND bicoterie.

(iii) f is positive and dual-minor $\Leftrightarrow$ ($MIN(f)$, $MIN(f^d)$) is an ND semicoterie.

As a corollary to the above, we can relate the numbers of ND coteries, ND bicoteries, and ND semicoteries to those of some classes of boolean functions.

Corollary 2.2: Let $|U| = n$.

(a) The number of ND coteries under U is equal to the number of positive, self-dual boolean functions of n variables.

(b) The number of ND bicoteries under U is equal to the number of positive boolean functions of n variables. (Note that (P, Q) and (Q, P) are two distinct bicoteries unless $P = Q$.)

(c) The number of ND semicoteries under U is equal to the number of positive, dual-minor boolean functions of n variables. This is the same as the number of coteries under U by Theorem 2.1. Also, as we shall see in the next section (Theorem 3.1), this is equal to the number of positive, self-dual functions of $n + 1$ variables. □

Remark 2.1: [18] contains the following asymptotic formula on the number, $\Psi(n)$, of positive boolean functions of n variables:

$$\Psi(n) \sim \ 2^{\binom{n}{n/2}} \exp\{\binom{n}{n/2-1}(\tfrac{1}{2^{n/2}} + \tfrac{n^2}{2^{n+5}} - \tfrac{n}{2^{n+6}})\}.$$

Reference [21] contains a lower bound

$$2^{\binom{n-1}{(n+1)/2}}$$

on the number of positive, self-dual boolean functions of n variables (for odd n). This bound is also found in [12] as a lower bound on the number of ND coteries. □

Remark 2.2: Given a set of subsets C, whether it is a coterie, i.e., f_C is dual-minor, can be decided in time polynomial in the number n of variables and the number of minimal vectors $m = \mid MIN(f_C) \mid$; this can be done by simply confirming that every pair of minimal vectors has a common variable with value 1. However, the complexity of deciding if f_C is dual-major,

and hence if f_C is self-dual, is not known, as pointed out in [12] in hypergraph setting. In fact, it is currently known as one of the open problems, which are possibly NP-complete.

The problem becomes solvable in polynomial time if we restrict ourselves to some special classes. For example, [10] contains some such special classes of hypergraphs (it also contains a variety of interesting applications of coterie-related concepts). Also, it can be shown that self-duality can be checked in polynomial time if and only if deciding whether two positive functions g and h are mutually dual can be done in polynomial time. The latter problem has a polynomial time algorithm if positive functions can be "dualized" in polynomial time. Some special classes of positive functions, including positive threshold functions, are known to have such polynomial-time dualization algorithms [4, 9, 23]. □

3 Construction of ND Coteries from Coteries

Let f be a positive boolean function of n variables, $x_1, x_2, \ldots, x_n$. The following decomposition, known as Shannon's decomposition, is a standard tool in switching theory:

$$f = f(x_n = 1)x_n \vee f(x_n = 0), \tag{1}$$

where $f(x_n = 0) \leq f(x_n = 1)$ since f is positive. Let $g = f(x_n = 0)$. g is thus a positive boolean function of $n-1$ variables, $x_1, x_2, \ldots, x_{n-1}$. If f is self-dual, then $f(x_n = 1) = f^d(x_n = 0)$[2] $= g^d$. Therefore, f can be decomposed as

$$f = g^d x_n \vee g. \tag{2}$$

Obviously, the converse (i.e., this f is self-dual) is also true, since

$$f^d = (g \vee x_n) \wedge g^d = gg^d \vee x_n g^d = f. \tag{3}$$

Also, as g is dual-minor by $f(x_n = 0) \leq f(x_n = 1)$, g defines a coterie by Theorem 2.1. From this analysis, the following theorem follows.

Theorem 3.1: There is a one-to-one correspondence between ND coteries under $U_n = \{u_1, u_2, \ldots, u_n\}$ and coteries under $U_{n-1} = \{u_1, u_2, \ldots, u_{n-1}\}$, where $n \geq 1$. In particular, the number of ND coteries under U_n is equal to the number of coteries (which is equal to the number of ND semicoteries by Corollary 2.2(c) in the previous section) under U_{n-1}. □

The above theorem slightly strengthens Lemma 4.1 and Theorem 4.1 in [12], since the correspondence here is one-to-one.

Given an arbitrary positive function g, $B = (MIN(g), MIN(g^d))$ is an ND bicoterie by Theorem 2.4. However, B is not necessarily a semicoterie. There are the following three cases.

(i) $g^d = g$: g is self-dual, which implies that $P = MIN(g)$ is an ND coterie and $B = (P, P)$ is an ND semicoterie.

[2] To evaluate $f^d(x_n = 0)$, first substitute $x_n = 0$ into $f(X)$ and then take the dual of the resulting function.

(ii) $g^d \neq g$ and g is dual-comparable: Since either g or g^d is dual-minor, assume without loss of generality that $g \leq g^d$ (i.e., g is dual-minor). Then g represents a coterie by Theorem 2.1, $B = (MIN(g), MIN(g^d))$ is an ND semicoterie by Theorem 2.4, and $f = g^d x_n \vee g$ represents an ND coterie by Theorem 2.2 and (3). However, g^d does not represent any coterie.

(iii) $g^d \neq g$ and g is not dual-comparable: Neither g nor g^d represents a coterie, and $B = (MIN(g), MIN(g^d))$ is not a semicoterie.

4 Tree Decomposition of ND Coteries

4.1 General Decomposition of Self-Dual Functions

We start with the following important theorem which states that the class of positive, self-dual (dual-minor, dual-major) functions is closed under compositions.

Theorem 4.1:

(a) Let f_i, $i = 1, 2, \ldots, k$, be positive and self-dual. Let $F(x_1, \ldots, x_k)$ be also positive and self-dual. Then

$$f = F(f_1, f_2, \ldots, f_k)$$

is positive and self-dual.

(b) Let f_i, $i = 1, 2, \ldots, k$, be positive and dual-minor (dual-major). Let $F(x_1, \ldots, x_k)$ be also positive and dual-minor (dual-major). Then

$$f = F(f_1, f_2, \ldots, f_k)$$

is positive and dual-minor (dual-major).

Proof. (a) The positivity of f is obvious and $f^d = f$ can be shown as follows.

$$[F(f_1, f_2, \ldots, f_k)]^d = F^d(f_1^d, f_2^d, \ldots, f_k^d)$$

$$= F(f_1, f_2, \ldots, f_k).$$

(b) Again, the positivity of f is obvious. We only prove the case where $\{f_i\}$ and F are dual-minor. The dual-major case can be analogously proved. Since $f_i \leq f_i^d$, we have

$$\forall X \in \{0,1\}^n : (f_1(X), f_2(X), \ldots, f_k(X)) \leq (f_1^d(X), f_2^d(X), \ldots, f_k^d(X)).$$

This, combined with the positivity of F and $F \leq F^d$, yields

$$F(f_1, f_2, \ldots, f_k) \leq F^d(f_1^d, f_2^d, \ldots, f_k^d).$$

□

Example 4.1: Consider the following four positive, self-dual functions.

$$\begin{aligned} F &= xy \vee yz \vee zx, \\ f_1 &= ab \vee bc \vee ca, \\ f_2 &= ac \vee cd \vee da, \\ f_3 &= e. \end{aligned}$$

After substitution and a routine simplification, one obtains

$$f = F(f_1, f_2, f_3) = ac \vee abd \vee abe \vee ade \vee bcd \vee bce \vee cde.$$

This f is positive and self-dual by Theorem 4.1(a). It is not difficult to see that

$$MIN(f) = \{\{a, c\}, \{a, b, d\}, \ldots, \{c, d, e\}\}$$

is in fact an ND coterie. □

Remark 4.1: It is known [12, 21] that there are very few positive, self-dual functions (hence ND coteries) that depend on[3] exactly n variables if $n = 1$, 2 or 3. In fact,

(i) $n = 1$: There is only one such function $f = x$, where $U = \{x\}$.

(ii) $n = 2$: None.

(iii) $n = 3$: There is only one such function $f = xy \vee yz \vee zx$, where $U = \{x, y, z\}$. □

We shall call the function $F(x, y, z) = xy \vee yz \vee zx$ the **basic majority function.** By the above remark, this function represents the simplest non-trivial ND coterie. Theorem 4.1 implies that any function composed of basic majority functions (without using constant 0 or 1) is positive and self-dual. An interesting question would be: what class of positive, self-dual boolean functions can be composed from copies of the basic majority function? We now define the **decomposable,** positive, self-dual function f on U as follows.

1. $f = x$, where $x \in U$, or
2. $f = F(f_1, f_2, f_3)$, where $F(x, y, z) = xy \vee yz \vee zx$, and f_1, f_2 and f_3 are decomposable functions on U.

We shall show in the next section that every ND coterie is decomposable. Thus the basic majority function is a "universal function" for positive, self-dual boolean functions, just like NAND and NOR functions are universal functions for all boolean functions. In fact, it turns out that every ND coterie has a more specific form of decomposition, called "tree decomposition", as will be discussed in the next section.

[3] f does not depend on a variable x if for any combination of the values of the other variables, f takes the same value regardless of the value of x.

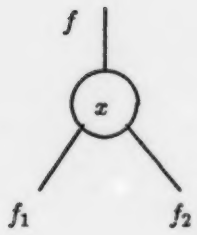

Figure 1: Tree decomposition of f.

4.2 Tree Decomposition of ND Coteries

Let $F(x,y,z) = xy \vee yz \vee zx$ (i.e., the basic majority function), and consider composition of f from f_1 and f_2 (or decomposition of f into f_1 and f_2) of the form,

$$f = F(f_1, f_2, x) = (f_1 \vee f_2)x \vee (f_1 \wedge f_2), \tag{4}$$

where f_1 and f_2 are positive, self-dual functions. Note that f_1 and f_2 may depend on x. By Theorem 4.1, f is positive and self-dual. By interpreting f as a family of subsets, we obtain the following result.

Theorem 4.2: Let C_1 and C_2 be ND coteries under U. The following set C is also an ND coterie.

$$C = M_I_N[\{G \cup \{x\} \mid G \in C_1 \cup C_2\} \cup \{G_1 \cup G_2 \mid G_1 \in C_1, G_2 \in C_2\}].$$

Here, $M_I_N[D]$, where D is a fimily of sets, denotes the family of sets obtained from D by eliminating duplicates and deleting all sets that properly contain another set in D.

Proof. Let $f_1 = f_{C_1}$ and $f_2 = f_{C_2}$. Then $C = MIN(f)$, where f is given by (4). The rest follows from point (i) given just after Theorem 2.4. □

The above decomposition (4) can be conveniently represented by a tree, as illustrated in Fig. 1, where the label of each non-leaf node indicates the variable x used for decomposition. By recursively applying this decomposition to the generated f_i's we may eventually end up with functions of single variables. A single-variable function is represented by a leaf node labeled by the variable. If all functions become single variables, then the generated finite binary tree is called the **decomposition tree**.

Example 4.2: Let f_1 and f_2 be defined by

$$f_1 = (x \vee y)u \vee xy \text{ and } f_2 = (u \vee x)v \vee ux.$$

The decomposition tree T for function f defined by

$$f = (f_1 \vee f_2)w \vee f_1 f_2 = (uy \vee xy \vee uv \vee vx)w \vee (ux \vee uvy \vee vxy)$$

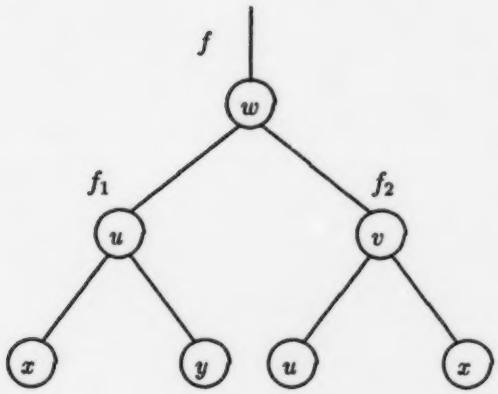

Figure 2: Illustration for Example 4.2.

is shown in Fig. 2. □

Remark 4.2: Different trees may represent the same ND coterie. For example, a tree consisting of the root labeled y and two child nodes labeled x and y represents an ND coterie $\{y\}$, which is also representable by a one-node tree labeled y. □

A positive, self-dual function f (or the corresponding ND coterie $MIN(f)$) is said to be **tree-decomposable** if

1. f is a single variable, or
2. f is composed of two tree-decomposable functions f_1 and f_2, using the composition rule (4).

In order to prove that every ND coterie is tree-decomposable, we start with some lemmas.

Lemma 4.1: Let a non-constant function g be positive and dual-major (resp., dual-minor). Then g can be represented by

$$g = f_1 \vee f_2 \vee \cdots \vee f_k \quad (\text{resp., } g = f_1 \wedge f_2 \wedge \cdots \wedge f_k) \tag{5}$$

for some $k > 0$, where each f_i is positive and self-dual.

Proof. We only prove the case where g is dual-major. The proof for the dual-minor case can be obtained by taking the dual of the dual-major case. Observe the following.

(i) Since g is dual-major, at least one of $g(X)$ and $g(\bar{X})$ equals 1 for any vector X (by Property 1.5(ii)).

(ii) Since g is positive, $g(Y) = 1$ implies $g(X) = 1$ for any $X \geq Y$. (Or equivalently, $g(Y) = 0$ implies $g(X) = 0$ for any $X \leq Y$.)

Given a dual-major g, we first present a formal procedure for constructing, for an arbitrary $Y \in MIN(g)$, a positive, self-dual function f_Y, satisfying $f_Y(Y) = 1$ and $f_Y \leq g$. The following procedure constructs f_Y in terms of its true domain and false domain, T and F, respectively (i.e., $f_Y(X) = 1$ (resp. 0) for all $X \in T$ (resp. F)).

1. Let $T := \{X \mid X \geq Y\}$ and $F := \{Z \mid g(Z) = 0\}$. [This makes sure that $f_Y(Y) = 1$ and $f_Y \leq g$ hold initially.]
2. If $\exists Z \notin T \cup F$ such that $\bar{Z} \in T$, then $F := F \cup \{X \mid X \leq Z\}$. [$f_Y(X)$ satisfies: if $f_Y(\bar{Z}) = 1$, then $f_Y(Z)$ must be 0, since it is self-dual.] Repeat this step as long as it is applicable.
3. If $T \cup F = \{0,1\}^n$, then stop. Otherwise, choose a $Z \notin T \cup F$ [$T \cup F$ contains neither Z nor $\bar{Z}$], and let $T := T \cup \{X \mid X \geq Z\}$. Return to step 2.

Clearly, the above procedure terminates, since T always enlarges in step 3. The fact that the resulting f_Y is indeed self-dual follows from the property of T and F that, for any vector X, exactly one of X and $\bar{X}$ belongs to T (and the other belongs to F). We now prove this assertion. Since $T \cup F = \{0,1\}^n$ upon termination, it suffices to show that neither T nor F contains both X and $\bar{X}$ during the construction. That this is true initially, i.e., for the sets T and F of step 1 above, can be shown as follows. Since $X \geq Y (\neq 0)$ obviously means $\bar{X} \not\geq Y$, $X \in T$ implies $\bar{X} \notin T$. Also, $X \in F$ implies $g(X) = 0$ and hence $g(\bar{X}) = 1$, i.e., $\bar{X} \notin F$, since g is dual-major. Next, consider the case where F is augmented in step 2. The resulting set F does not contain both X and $\bar{X}$ for any X, since for any new vector $X \leq Z$ in F, $\bar{X} \in T$ holds because $\bar{X} \geq \bar{Z}$. Finally, in step 3, the resulting set T has this property, since for any new vector $X \geq Z$ in T, $\bar{X} \notin T$ holds because $\bar{X} \leq \bar{Z}$ and $\bar{Z} \notin T$ (if $\bar{Z} \in T$ then $Z \in F$ by step 2). The positivity of f_Y is obvious from the construction.

Applying the above procedure to every minimal vector Y of g, we obtain

$$g = \bigvee_{Y \in MIN(g)} f_Y.$$

□

Remark 4.3: In the above construction, many f_Y can be redundant. It is usually possible to make k in (5) fairly small. However, the true minimization is left open. □

Lemma 4.2: Let f be a positive, self-dual function on U. Then f can be decomposed as follows.

$$f = (f_1 \vee f_2 \vee \cdots \vee f_k)x \vee (f_1 \wedge f_2 \wedge \cdots \wedge f_k),$$

where $x \in U$ and each f_i is a positive, self-dual function, which does not depend on x.

Proof. Take any $x \in U$ and decompose f with respect to x as in (2) of Section 3, i.e.,

$$f = f(x = 1)x \vee f(x = 0) = g^d x \vee g,$$

where $g = f(x = 0)$ and g is dual-minor. By Lemma 4.1, g^d can be represented as

$$g^d = f_1 \vee f_2 \vee \cdots \vee f_k$$

and g as

$$g = f_1 \wedge f_2 \wedge \cdots \wedge f_k$$

for some self-dual functions, $f_1, f_2, \ldots, f_k$, on $U - \{x\}$. □

The next lemma shows that f can be composed from functions $f_1, f_2, \ldots, f_k$, using only the basic majority functions.

Lemma 4.3: Consider the decomposition

$$f = (f_1 \vee f_2 \vee \cdots \vee f_k)x \vee (f_1 \wedge f_2 \wedge \cdots \wedge f_k)$$

in Lemma 4.2. This can be represented as

$$f = (f_1 \vee F_2)x \vee (f_1 \wedge F_2)$$

and

$$F_i = (f_i \vee F_{i+1})x \vee (f_i \wedge F_{i+1})$$

for $i = 2, 3, \ldots, k-1$, where $F_k = f_k$.

Proof. Easy if one treats the two cases, $x = 0$ and $x = 1$, separately. □

Theorem 4.3: Any ND coterie is tree-decomposable.

Proof. Apply Lemmas 4.2 and 4.3 repeatedly until all f_i become single variables. □

Corollary 4.1: Any ND coterie is decomposable. □

Now, tree-decomposition discussed so far defines a single-output circuit made up of copies of the basic majority function whose inputs are variables from U (no constants) and whose output is f (Fig. 3).

To check whether a given subset $S \subseteq U$ contains a quorum of the corresponding ND coterie, we can simply feed (the vector representing) S into this circuit. If the output is 1, then S contains a quorum of the ND coterie; otherwise, it does not. In this sense, the circuit is a compact representation of an ND coterie, which often requires less space than a list of all the quorums of the ND coterie.

Remark 4.4: Tree decomposition can be generalized if we substitute the basic majority function of (4) by any positive self-dual $(k+1)$-ary function F and represent

$$F(f_1, f_2, \ldots, f_k, x)$$

by a node that is labeled by x and has $k(\geq 2)$ child nodes, $f_1, f_2, \ldots, f_k$. In particular, if F represents a vote-assignable ND coterie (see Section 6 for definition) in which each f_i has weight 1 and x has weight $k-1$, it is given by

$$f = (f_1 \vee f_2 \vee \cdots \vee f_k)x \vee (f_1 \wedge f_2 \wedge \cdots \wedge f_k), \tag{6}$$

and may be considered as a direct extension of (4). □

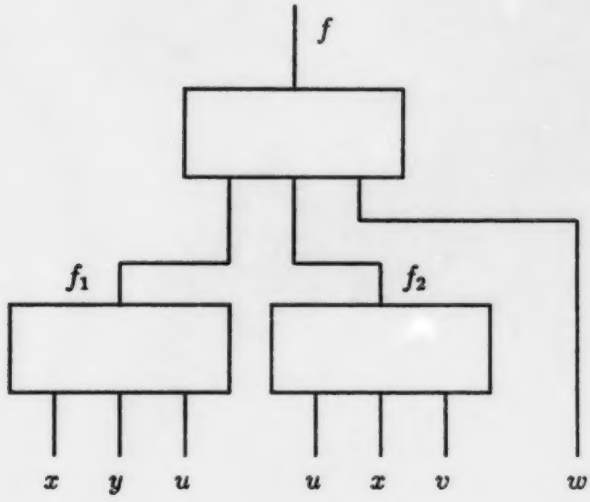

Figure 3: A network of basic majority functions corresponding to Fig. 2.

4.3 Non-redundantly Decomposable ND Coteries

An ND coterie is said to be **non-redundantly decomposable** if it has a tree representation in which all node labels are different. In this case, we need to consider a decomposition

$$f = (f_1 \vee f_2)y \vee (f_1 \wedge f_2), \tag{7}$$

where both f_1 and f_2 are positive, self-dual and, furthermore, their variable sets are mutually disjoint and do not contain y. In decomposition (7), we say that y is the root of the corresponding decomposition tree.

Lemma 4.4: Assume that a positive and self-dual function f has a decomposition of the form (7) and that f_1 depends on a variable w. Then f has another decomposition tree with root w if and only if $f_1 = w$.

Proof. If: If $f_1 = w$, we have

$$f = (w \vee f_2)y \vee wf_2 = (y \vee f_2)w \vee yf_2.$$

Therefore, f has a decomposition tree whose root is w.

Only-if: Assume that f_1 in decomposition (7) depends on variable w and some others. Assume that w is the root of some decomposition tree for f. Using a theorem of boolean algebra, we derive

$$\begin{aligned} f &= (f_1 \vee f_2)y \vee (f_1 \wedge f_2) \\ &= (f_1(w=1) \vee f_2)yw \vee (f_1(w=1) \wedge f_2)w \end{aligned}$$

$$\begin{aligned} & \vee(f_1(w=0) \vee f_2)y \vee (f_1(w=0) \wedge f_2) \\ = \; & [(f_1(w=1) \vee f_2)y \vee (f_1(w=1) \wedge f_2)]w \\ & \vee(f_1(w=0) \vee f_2)y \vee (f_1(w=0) \wedge f_2). \end{aligned}$$

Define g by

$$g^d = (f_1(w=1) \vee f_2)y \vee (f_1(w=1) \wedge f_2),$$

i.e., the coefficient of w in the expression for f. Since w is the root, g^d must be of the form

$$g^d = g_1 \vee g_2,$$

where g_1 and g_2 are positive, self-dual functions defined on disjoint sets of variables. Now, as noted in Remark 1.1, a positive function has a unique disjunctive form, called the minimum sum, consisting of all of its prime implicants. ¿From the definition of g^d, it is easy to see that the minimum sum of g^d consists of prime implicants of the form

$$yP_i \text{ or } P_1P_2,$$

where P_1 (resp. P_2) is a prime implicant of $f_1(w=1)$ (resp. f_2). Since both g_1 and g_2 cannot depend on y, assume without loss of generality that all terms of the form yP_i belong to g_1. Then g_2 must contain at least one term P_1P_2, and g_1 and g_2 share all the variables in P_1, a contradiction to the assumption on $\{g_j\}$. □

Lemma 4.5: Let f be a positive, self-dual function that depends on more than three variables[4] and be non-redundantly decomposable. A variable y can be the root of a decomposition tree for f, if and only if it appears in every subset $G \in MIN(f)$ of the smallest size.

Example 4.3: Consider the positive, self-dual function f of Example 4.2,

$$f = ux \vee wxy \vee vwx \vee vxy \vee uvw \vee uvy \vee uwy.$$

The above theorem asserts that either one of u and x (and no other variable) can be the root of a decomposition tree. □

Proof of Lemma 4.5. Consider decomposition (7), where y is the root. First, assume that both f_1 and f_2 depend on more than one variable. In this case, by Lemma 4.4, only y can be the root of a decomposition tree. By Remark 4.1, the assumption means that both f_1 and f_2 depend on at least three variables, and that each $J \in MIN(f_1 \vee f_2)$ satisfies $|J| \geq 2$. (Otherwise, f_1 or $f_2 = x$ for some variable x.) Thus, Theorem 4.2 implies that a subset $G \in MIN(f)$ has the smallest size in $MIN(f)$ if and only if $G = H \cup \{y\}$ for some $H \in MIN(f_1 \vee f_2)$ of the smallest size in $MIN(f_1 \vee f_2)$, since any $I \in MIN(f_1 \wedge f_2)$ has size at least $|H|+2$. Therefore, y appears in every $G \in MIN(f)$ of the smallest size.

Next, assume that f_1 or f_2 depends on exactly one variable. Without loss of generality, let f_1 depend on $w(\neq y)$ only. Then any $G \in MIN(f)$ of the smallest size, which is of the

[4]Recall that positive, self-dual functions of up to three variables are very simple, as commented in Remark 4.1.

form $H \cup \{y\}$ for some $H \in MIN(f_1 \vee f_2)$, has size 2. Any subset $I \in MIN(f_1 \wedge f_2)$ has size > 2, since, otherwise, f_2 would also depend on only one variable and f would be equal to the basic majority function, which is ruled out by the assumption of the lemma. This proves that y appears in every $G \in MIN(f)$ of the smallest size (size 2). Since f_2 depends on more than one variable, $G \in MIN(f)$ has the smallest size (size 2) if and only if $G = \{w, y\}$. In other words, w also appears in every $G \in MIN(f)$ of the smallest size (size 2). In this case, as shown in Lemma 4.4, w can also be the root of a decomposition tree for f. □

Based on Lemma 4.5, it can be decided whether or not a given positive function f represents a non-redundantly decomposable ND coterie as follows.

Algorithm DECOMPOSE:

1. If f depends on at most three variables, then f represents a non-redundantly decomposable ND coterie if and only if $f = x$ for some variable x or f is the basic majority function. Stop. If f depends on more than three variables, go to step 2.
2. Check if f is dual-minor (i.e., $MIN(f)$ satisfies the intersection property). If not, f does not represent a coterie. Stop. Otherwise, check if there exists a variable that appears in all subsets $G \in MIN(f)$ of the smallest size. If not, f does not represent a non-redundantly decomposable ND coterie. Stop. Otherwise, take one such variable y and go to step 3.
3. Compute $f(y = 1)$ (i.e., $S_1 = MIN\{G - \{y\} \mid G \in MIN(f)\}$). Decompose $f(y = 1)$ into disjunction of two positive, dual-minor functions, f_1 and f_2, having disjoint sets of variables as follows. Starting with a subset in family S_1, form the largest subfamily of S_1 by successively choosing subsets from S_1 which intersect some subset(s) already chosen. Let f_1 represent this subfamily and let f_2 represent the subfamily of remaining subsets. Then check if $f(y = 0)$ (i.e., $S_0 = \{G \mid G \in MIN(f) \text{ and } y \notin G\}$) satisfies
$$f(y = 0) = f_1 \wedge f_2. \tag{8}$$
If not, f does not represent a non-redundantly decomposable ND coterie. Stop. Otherwise, go to step 4.
4. Apply steps 1 to 3 recursively to test if both f_1 and f_2 represent non-redundantly decomposable ND coteries. f represents a non-redundantly decomposable ND coterie if and only if both of them represent non-redundantly decomposable ND coteries. □

Example 4.4: Let us apply DECOMPOSE to the positive, self-dual function, f, of Example 4.3. In step 1, we see that each of u and x can be the root. Choosing x as the root in step 2, we obtain
$$f(x = 1) = (vw \vee vy \vee wy) \vee u,$$
i.e., $f_1 = vw \vee vy \vee wy$ and $f_2 = u$. Since f_1 is the basic majority function and f_2 is a single variable, this gives the non-redundant tree-decomposition shown in Fig. 4. □

We show below that condition (8) in step 3 of DECOMPOSE implies that f is self-dual under the assumption that f_1 and f_2 are self-dual. Since $f(y = 1) = f_1 \vee f_2$, we have
$$(f(y = 1))^d = (f_1 \vee f_2)^d = f_1^d \wedge f_2^d = f_1 \wedge f_2.$$

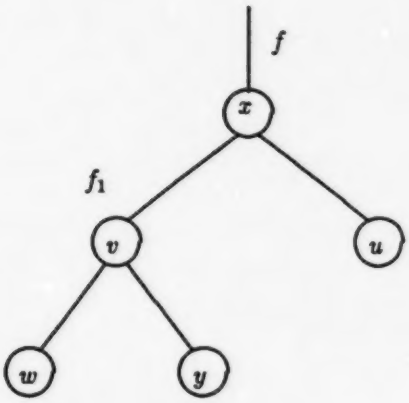

Figure 4: Non-redundant tree decomposition of Examples 4.3 and 4.4.

Therefore,

$$f = f(y = 1)y \vee f(y = 0) \quad \text{(see (1))}$$

becomes self-dual (see (2)) if and only if

$$f = f(y = 1)y \vee (f(y = 1))^d,$$

i.e., $f(y = 0) = f_1 \wedge f_2$ holds.

We now show that the time complexity of DECOMPOSE is polynomial. It is clear that steps 1 and 2 can be carried out in polynomial time in the number n of variables and $m = |MIN(f)|$. To see that step 3 is a polynomial time computation, first note that, by definition, S_1 and S_0 can be constructed from $MIN(f)$ in polynomial time. Decomposition of f into f_1 and f_2 can also be done in polynomial time, just by following the procedure stated therein. Finally, condition (8) can be tested in polynomial time by directly checking if $MIN(f(y = 0)) = MIN(f_1 \wedge f_2)$, where $MIN(f_1 \wedge f_2)$ can be obtained in polynomial time, using the relation

$$MIN(f_1 \wedge f_2) = MIN\{(G_1 \cup G_2) \mid G_1 \in MIN(f_1), G_2 \in MIN(f_2)\}.$$

Step 4 of DECOMPOSE applies steps 1-3 recursively to f_1 and f_2. Since each decomposition of f into f_1 and f_2 partitions the set of variables into two disjoint sets, the entire process can clearly be completed in polynomial time.

Theorem 4.4: Given a positive function f in the form of $MIN(f)$ (i.e., as a set of its prime implicants), it can be decided in time polynomial in the number of variables n and $|MIN(f)|$ whether f represents a non-redundantly decomposable ND coterie. □

Some positive, self-dual functions are not non-redundantly decomposable. For example, the ND coterie consisting of all the 3-element subsets of $U = \{u_1, u_2, \ldots u_5\}$ gives rise to such a function.

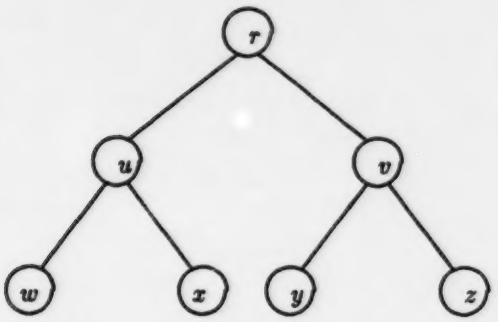

Figure 5: A labeled complete binary tree used in Agrawal-El Abbadi construction.

Remark 4.5: The class of non-redundantly decomposable coteries proposed by Agrawal and El Abbadi [1] is constructed by the following algorithm. The input to the algorithm is a complete binary tree, in which all nodes have distinct labels.

Algorithm GENERATE:

1. Start with the set $\{r\}$, where r is the root. r is *unscanned.*
2. If no set contains an unscanned non-leaf node, then halt. The current family of sets is an ND coterie. Otherwise, choose an arbitrary set S containing an unscanned non-leaf node x, and go to step 3.
3. Let y and z be the child nodes of x. Generate from S the following three sets: $S \cup \{y\}$, $S \cup \{z\}$, and $(S \cup \{y,z\}) - \{x\}$. x is now scanned but y and z are not. Return to step 2.□

As an example, consider the complete binary tree given in Fig. 5. Algorithm GENERATE generates sets S as illustrated in the following table. Read it from left to right. The underlined labels indicate those nodes which are neither scanned nor are leaf nodes. Therefore, the 15 sets containing only labels without underlines constitute the constructed coterie.

{r}	{r,u}	{r,u,w}, {r,u,x}, {r,w,x}	
	{r,v}	{r,v,y}, {r,v,z}, {r,y,z}	
	{u,v}	{u,w,v}	{u,w,v,y}, {u,w,v,z}, {u,w,y,z}
		{u,x,v}	{u,x,v,y}, {u,x,v,z}, {u,x,y,z}
		{w,x,v}	{w,x,v,y}, {w,x,v,z}, {w,x,y,z}

In view of the definition of the tree decomposition and Theorem 4.2, it is not difficult to see that the coterie constructed as above is qual to $MIN(f)$, where f is the boolean function associated with the given complete binary tree. The condition that the tree be complete can be easily incorporated into our algorithm DECOMPOSE by testing in step 2 whether two disjoint subsets of variables of f_1 and f_2 have the same cardinality.

It is clear that algorithm GENERATE can be applied not only to a complete binary tree but also to any binary tree with distinct node labels. Given such a tree as input, GENERATE will generate the non-redundantly decomposable ND coterie associated with the tree. GENERATE can also be generalized to non-binary trees with distinct node labels, in which the boolean function associated with each node is of type (6) stated in Remark 4.4. In this case, the family of sets generated from S in step 3 of GENERATE should read

$$S \cup \{y_1\}, S \cup \{y_2\}, \ldots, S \cup \{y_k\}, (S \cup \{y_1, y_2, \ldots, y_k\}) - \{x\},$$

if the current node has label x and k child nodes $y_1, y_2, \ldots, y_k$. The generalization of this type was already mentioned in [1]. □

Remark 4.6: The number of labeled rooted trees on n nodes, in which each node has either 0 or 2 child nodes, gives an upper bound on the number of non-redundantly decomposable ND coteries, as different trees may be representable by the same boolean function (see Remark 4.2). Such trees exist only for odd $n = 2m + 1$, and their number is given as follows (see problem 21 in Section 3.4.4 of [17]).

$$\binom{2m+1}{m}(2m)!/2^m.$$

This is asymptotically much smaller than the number of ND coteries cited in Remark 2.1. □

4.4 Tree Decomposition of General Coteries

Tree decomposition of ND coteries discussed in Section 4.2 can be generalized to all coteries with a slight modification. This will reveal an interesting structural relation between a coterie and the ND coteries that dominate it.

Consider the basic majority function, $F(x, y, z) = xy \vee yz \vee zx$. It is easily seen that $F(x, y, 0) = xy$ (AND function) and $F(x, y, 1) = x \vee y$ (OR function), which are conveniently represented by nodes with labels 0 and 1, respectively. As noted before, xy is dual-minor and $x \vee y$ is dual-major.

Now, take a tree decomposition as discussed in Section 4.2, with the modification that some nodes may now have labels 0 (but not 1). Since every node in such a tree represents a positive, dual-minor function (recall that a self-dual function is also dual-minor), repeated applications of Theorem 4.1(b) show that the function f it represents is positive and dual-minor, i.e., $C = MIN(f)$ is a coterie.

Example 4.5: The tree shown in Fig. 6 represents

$$\begin{aligned} f &= (f_1 \vee f_2)w \vee f_1 f_2 \\ &= (uv \vee ux \vee vx \vee xy)w \vee uxy \vee vxy, \end{aligned}$$

where $f_1 = xy$ and $f_2 = uv \vee vx \vee xu$. It is easy to see that $MIN(f)$ is in fact a coterie. □

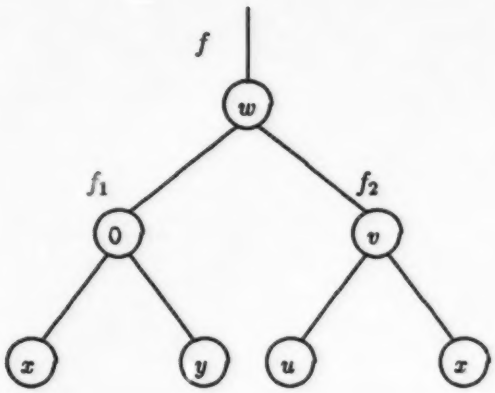

Figure 6: A tree decomposition with basic majority functions and AND function.

The next theorem says that the above decomposition is general, i.e., any coterie has such a decomposition.

Theorem 4.5:

(a) A function, represented by a tree composed of copies of the basic majority function and/or AND function, is positive and dual-minor.

(b) Any positive, dual-minor function f has a tree decomposition composed of copies of the basic majority function and/or AND function.

Proof. As (a) was already shown, we prove (b) only. Since f is positive and dual-minor, Lemma 4.1 states that f can be represented as

$$f = f_1 \wedge f_2 \wedge \cdots \wedge f_k$$

for some $k > 0$, where each f_i is positive and self-dual. Recall that each f_i has a tree decomposition into basic majority functions by Theorem 4.3, and the conjunction of k f_i's in the above expression can be realized by using $k-1$ AND functions $F(x, y, 0)$ as follows.

$$f_1 \wedge f_2 \wedge \cdots \wedge f_k = (\cdots((f_1 \wedge f_2) \wedge f_3) \cdots \wedge f_k).$$

Therefore, f has a tree decomposition as stated. □

Assume now that a tree decomposition T of the above theorem is given for a positive dual-minor function f. Since the basic majority function is self-dual and the dual of AND is OR, i.e.,

$$F^d(x, y, 0) = F(x, y, 1),$$

the tree decomposition T^d, obtained by switching all labels 0 to 1, represents f^d. (See footnote 2 for the interpretation of $F^d(x,y,0)$.) Therefore, this can be used as a convenient way of defining an ND semicoterie,

$$B = (MIN(f), MIN(f^d))$$

as well as its representation by two trees.

A more interesting observation is that a self-dual function can be defined from T if all labels 0 in T are replaced by some variables. Since $F(x,y,0) \leq F(x,y,u)$ for any variable u (which is possibly equal to x or y), the resulting self-dual function g satisfies $f \leq g$. As observed in Section 2, this means that g represents an ND coterie that dominates the original coterie $MIN(f)$. By assigning different variables to labels 0, we can generate many ND coteries that dominate coterie $MIN(f)$.

As an example, note that the tree decomposition of Fig. 2 can be obtained by assigning variable u to the node with label 0 in Fig. 6. It is easy to check that the ND coterie of Fig. 2, given in Example 4.2, dominates the coterie of Fig. 6, given in Example 4.5.

5 Decomposition of ND Coteries into Smaller Functions

Motivated by the recent work by Neilsen and Mizuno [22], we investigate in this section a special case of the general decomposition in Theorem 4.1(a).

5.1 Expansion Using a Self-Dual Function

Consider a function of the form

$$f = F(h, x_1, x_2, \ldots, x_n), \tag{9}$$

where F and h are positive and self-dual. We rewrite f as

$$\begin{aligned} f &= F(x_0, x_1, x_2, \ldots, x_n), \\ x_0 &= h(x_1, x_2, \ldots, x_n), \end{aligned}$$

and expand F on x_0 as in (1). We thus obtain

$$\begin{aligned} f(X) &= F(1,X)x_0 \vee F(0,X) \\ &= g^d(X)x_0 \vee g(X) \\ &\qquad \text{(by (2) since } f \text{ is self dual by Theorem 4.1(a))} \\ &= g^d(X)h(X) \vee g(X), \end{aligned} \tag{10}$$

where $X = (x_1, x_2, \ldots, x_n)$, $g(X) = F(0,X)$, and g is dual-minor. This decomposition may be considered as a generalization of (2) in the sense that positive, self-dual "function" x_0 is replaced by a general positive, self-dual function h.

Decomposition (10) is useful if g and h are constructed from smaller functions; more precisely, if g depends on less than $n-1$ variables (i.e., F depends on less than n variables) and h is composed of a known class of functions, each depending on less than n variables. In the following subsections, we shall discuss two cases that meet this requirement.

5.2 Decomposition into Positive Self-Dual and Positive Functions

Before presenting the main result of this subsection, we describe some properties of positive functions $f(X)$, where $X = (x_1, x_2, \ldots, x_n)$ and $n \geq 3$. Expanding f on x_{n-1} and x_n by (1), we obtain

$$f(X) = f_{11}(X')x_{n-1}x_n \vee f_{10}(X')x_{n-1} \vee f_{01}(X')x_n \vee f_{00}(X'), \tag{11}$$

where

$$X' = (x_1, x_2, \ldots, x_{n-2}) \text{ and}$$

$$f_{ab}(X') = f(X', a, b) \text{ for } a = 0, 1 \text{ and } b = 0, 1.$$

By the positivity of f, $\{f_{ab}\}$ are all positive, and we have

$$f_{00} \leq f_{10} \leq f_{11} \text{ and } f_{00} \leq f_{01} \leq f_{11}.$$

Note that f_{10} and f_{01} are, in general, not comparable.

Lemma 5.1: Consider decomposition (11) of a positive function $f(X)$ of n variables, where $n \geq 3$. f is self-dual if and only if

$$f_{00}(X') = f_{11}^d(X') \text{ (equivalently, } f_{00}^d(X') = f_{11}(X'))$$

$$f_{01}(X') = f_{10}^d(X') \text{ (equivalently, } f_{01}^d(X') = f_{10}(X')). \tag{12}$$

Proof.

$$f^d = (f_{11}^d \vee x_{n-1} \vee x_n)(f_{10}^d \vee x_{n-1})(f_{01}^d \vee x_n)f_{00}^d \text{ (from (11))}.$$

Since $f_{00} \leq f_{10} \leq f_{11}$ implies $f_{00}^d \geq f_{10}^d \geq f_{11}^d$, the above expression can be simplified to

$$f^d = f_{00}^d x_{n-1}x_n \vee f_{01}^d x_{n-1} \vee f_{10}^d x_n \vee f_{11}^d.$$

By considering all possible assignments to x_{n-1} and x_n, and by noting that $f_{00} = f_{11}^d$ implies $f_{00}^d = f_{11}$ and so forth, it follows that $f = f^d$ holds if and only if condition (12) holds. □

Define $p(X')$ and $q(X')$ as follows.

$$p^d(X') = f_{11}(X') \text{ and } q(X') = f_{10}(X').$$

Lemma 5.1 implies that a positive function f is self-dual if and only if it has a representation

$$f(X) = p^d(X')x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n \vee p(X'), \tag{13}$$

such that $p(X')$ and $q(X')$ are positive functions satisfying

$$p(X') \leq q(X') \leq p^d(X') \tag{14}$$
$$\text{(i.e., } p(X') \leq q^d(X') \leq p^d(X').)$$

Note that (13) can be considered as a rule for constructing a self-dual function f from smaller functions, where $p(X')$ and $q(X')$ can be chosen independently as long as (14) is satisfied. A special case of this decomposition is

$$f(X) = x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n \tag{15}$$

(i.e., $p^d(X') = 1$ and $p(X') = 0$). In this case, condition (14) is trivially satisfied for any positive function $q(X')$.

Remark 5.1: Relation (15) implies the following inequality:

(The number of positive self-dual functions of up to n variables) $\geq$
(The number of positive functions of up to $n-2$ variables).

However, this does not give any better lower bound on the number of positive self-dual functions than the one given in Remark 2.1. □

Now take any positive self-dual function $f(X)$ and consider its decomposition (13). It can be rewritten as follows.

$$\begin{aligned} f(X) &= p^d(X')x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n \vee p(X') \\ &= p^d(X')(x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n) \vee p(X') \quad \text{(by (14))} \\ &= g^d(X)h(X) \vee g(X) \end{aligned}$$

where

$$g(X) = p(X'), \text{ and}$$
$$h(X) = x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n, \tag{16}$$

proving the next theorem.

Theorem 5.1: Decomposition (10) is universal in the sense that any positive, self-dual function f of n variables can be composed from a positive function q (given in (16)) of $n-2$ variables and a positive, self-dual function of $n-1$ variables,

$$p^d(X')y \vee p(X'). \tag{17}$$

□

5.3 Decomposition into Disjoint Sets of Variables

Another interesting case is where the set of variables can be decomposed into two disjoint sets. In this case, (10) becomes

$$f(X) = g^d(Z)h(Y) \vee g(Z), \tag{18}$$

where

$$\begin{aligned} X &= (x_1, x_2, \ldots, x_n), \\ Y &= (x_{i_1}, x_{i_2}, \ldots, x_{i_k}), \\ Z &= (x_{i_{k+1}}, x_{i_{k+2}}, \ldots, x_{i_n}), \end{aligned} \tag{19}$$

$(x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ is a permutation of $(x_1, x_2, \ldots, x_n)$.

Note that decomposition (2) is a special case of (18), where $h(Y)$ is a single variable. Decomposition (18) is not universal, since, for example, the ND coterie consisting of all the 3-element subsets of $U = \{u_1, u_2, \ldots, u_5\}$ does not have such decomposition.

Remark 5.2: In [22], Neilsen and Mizuno have proposed a coterie join algorithm that constructs an ND coterie C_3 over ground set $U'' = (U - \{x_0\}) \cup U'$ from two ND coteries C_1 and C_2 under disjoint ground sets U and U', respectively, where x_0 is a designated element in U. That is, $G_3 \in C_3$ holds if and only if

$$G_3 = G_1 \text{ for some } G_1 \in U \text{ with } x_0 \notin G_1, \text{ or}$$
$$G_3 = (G_1 - \{x_0\}) \cup G_2 \text{ for some } G_2 \in C_2 \text{ and some } G_1 \in U \text{ with } x_0 \in G_1.$$

They show, among other things, that C_3 is an ND coterie if and only if so are C_1 and C_2.

Now, let $F(z_0, Z)$ (resp. $h(Y)$) be the positive self-dual function associated with C_1 (resp. C_2) given in the following form

$$F(z_0, Z) = g^d(Z) z_0 \vee g(Z),$$

where

$$g^d(Z) = F(1, Z) \text{ and } g(Z) = F(0, Z).$$

Then it is not difficult to see that the ND coterie given by $f(X)$ of (18) is nothing but the above ND coterie C_3. In this sense, (18) is a boolean description of their join algorithm. An advantage of the boolean approach is that the NDness is an immediate consequence of the self-duality of f, though a rather involved argument is necessary to show it directly from the join algorithm.

Moreover, the construction of C_3 as above is valid even if Y and Z are not disjoint, as it can be considered as an algorithm to compute $MIN(f)$ of general decomposition (10). However, in this case, some generated sets may be included in other sets, and, therefore, operation $M_I_N[C_3]$ is necessary to obtain the corresponding ND coterie. □

We point out here the relationship of decomposition (18) to the classical decomposition of Ashenhurst [2] (referred to henceforth as **A-decomposition**), which can be applied to any boolean function. A boolean function $f(X)$ of n variables with $n \geq 3$ has A-decomposition if it can be represented as

$$f(X) = F(h(Y), Z), \tag{20}$$

where F is a function of $n - k + 1$ variables, h is a function of k variables with $k \geq 2$, and Y and Z are given by (19). The next lemma shows that decomposition (18) is equivalent to A-decomposition applied to the functions representing ND coteries.

Lemma 5.2: Let a boolean function $f(X)$ that depends on n variables with $n \geq 3$ be positive and self-dual. Then f has decomposition of the form (18) if and only if it has A-decomposition of the form (20).

Proof. Decomposition (18) is obviously a special case of A-decomposition. Conversely, if f has A-decomposition, it can be decomposed as follows.

$$f(Y,Z) = F(1,Z)h(Y) \vee F(0,Z)\bar{h}(Y). \tag{21}$$

Since f depends on Y, there is an assignment $Z = A$ such that $F(1,A) = 1$ and $F(0,A) = 0$. (If not, there is an assignment such that $F(0,A) = 1$ and $F(1,A) = 0$. In this case, interchange the roles of h and $\bar{h}$ by redefining $\bar{h}$ as h.) Then

$$f(Y,A) = h(Y),$$

and h is positive since f is positive. Next, consider assignments $Y = B_1$ and $Y = B_2$ such that $B_1 > B_2$, and $h(B_1) = 1, h(B_2) = 0$, that is $f(B_1,Z) = F(1,Z)$ and $f(B_2,Z) = F(0,Z)$. Since f is positive in Y, this implies that

$$F(1,Z) \geq F(0,Z),$$

and (21) reduces to

$$f(Y,Z) = F(1,Z)h(Y) \vee F(0,Z). \tag{22}$$

Now,

$$\begin{aligned} f^d(Y,Z) &= (F^d(1,Z) \vee h^d(Y))F^d(0,Z) \\ &= F^d(0,Z)h^d(Y) \vee F^d(1,Z), \end{aligned}$$

and since f is self-dual,

$$F^d(0,Z)h^d(Y) \vee F^d(1,Z) = F(1,Z)h(Y) \vee F(0,Z).$$

Choosing an assignment (e.g., $Y = 0$) to Y such that $h(Y) = h^d(Y) = 0$, we have

$$F(0,Z) = F^d(1,Z) \text{ (i.e., } F(1,Z) = F^d(0,Z)),$$

and choosing an assignment to Z such that $F(1,Z) = 1$ and $F(0,Z) = 0$, we have

$$h(Y) = h^d(Y) \text{ (i.e., } h \text{ is self dual).}$$

Therefore, (22), hence (21) is a decomposition of the form (18). □

Remark 5.3: Properties and algorithms related to A-decomposition have been extensively studied in the literature of boolean functions (e.g., [2, 14]). Interestingly, this subject has also been studied in other areas under various names, such as committees in game theory [25], clutters in set systems [5], and modules in coherent sytems of reliability theory [6]. Based on the result of Billera [5], Möhring [20] developed an algorithm to test the existence of A-decomposition for a positive boolean function (in boolean terminology), which runs in polynomial time in n and $m = |MIN(f)|$. The time complexity was further improved to $O(n^2m^3)$ in [24]. □

As a special case of (18), assume that

$$g^d(Z) = g'(Z') \vee z \quad \text{and} \quad g(Z) = g'(Z') \wedge z, \tag{23}$$

where z is a variable in Z, and Z' denotes the vector of variables Z from which z is deleted, and

$$g'(Z') \text{ is positive and self-dual.}$$

Then (18) becomes

$$f(X) = (g'(Z') \vee h(Y))z \vee (g'(Z') \wedge h(Y)),$$

which is in the same form as (7). Therefore, non-redundantly decomposable ND coteries in Section 4.3 are contained in the class of ND coteries which are obtained by recursively applying decomposition (18) to all the generated functions (as long as they contain more than 3 variables).

6 Vote-Assignable Coteries

6.1 Vote-Assignable Coteries and Threshold Functions

A boolean function $f(X)$ is called a **threshold function** [21] if there is a set of **weights**, $w_1, \ldots, w_n$, and a **threshold** T such that $f(X) = 1$ if and only if $\sum_i w_i x_i \geq T$. Here, all weights and thresholds are nonnegative integers. Clearly, the basic majority function is a threshold function with weights $w_1 = w_2 = w_3 = 1$ and threshold $T = 2$.

Since $f^d(X) = \bar{f}(\bar{X})$ by definition, we have

$$\begin{aligned} f^d(X) = 1 &\Leftrightarrow f(\bar{X}) = 0 \\ &\Leftrightarrow \textstyle\sum_i w_i(1 - x_i) < T \\ &\Leftrightarrow \textstyle\sum_i w_i x_i \geq \sum_i w_i + 1 - T. \end{aligned}$$

From this observation, it follows that

(i) If f is a threshold function, then f^d is also a threshold function, which is realizable with the same set of weights as f and a threshold of $T^d = W + 1 - T$, where $W = \sum_i w_i$,

(ii) A threshold function f is dual-comparable, and is dual-minor (-major) if $T \geq T^d$ ($T \leq T^d$), and

(iii) A threshold function f is self-dual if $W = \sum_i w_i$ is odd and $T = (W + 1)/2$.

A coterie represented by f is said to be **vote-assignable** [12] if there is a set of non-negative integer weights, $w_1, \ldots, w_n$, such that $W = \sum_i w_i$ is odd and $f(X) = 1$ if and only if $\sum_i w_i x_i \geq (W + 1)/2$. A bicoterie (P, Q) is said to be **vote-assignable** [11] if both f_P and f_Q are positive threshold functions. The following results now easily follow from the above three facts [11, 12].

Theorem 6.1:

(a) $MIN(f)$ is a vote-assignable ND coterie if and only if f is a positive, self-dual threshold function.

(b) Let $B = (MIN(g), MIN(g^d))$. B is a vote-assignable ND bicoterie if and only if g is a positive threshold function.

(c) Let $B = (MIN(g), MIN(g^d))$. B is a vote-assignable ND semicoterie if and only if g is a positive, dual-minor threshold function. □

Theorem 6.2: Let $B = (P, Q)$ be a vote-assignable ND bicoterie. Then either B or $B' = (Q, P)$ is a vote-assignable semicoterie.

Proof. Follows from fact (ii) above, Theorem 2.4, and Theorem 6.1(b). □

Theorem 6.3: Let f represent any vote-assignable ND coterie under $U_n = \{u_1, u_2, \ldots, u_n\}$. f can be constructed from a vote-assignable ND semicoterie $B = (MIN(g), MIN(g^d))$ under $U_{n-1} = \{u_1, u_2, \ldots, u_{n-1}\}$ by $f = g^d x_n \vee g$. Moreover, the correspondence between g and f is one-to-one. □

Corollary 6.1: Let $|U| = n$.

(a) The number of vote-assignable ND coteries under U is equal to the number of positive, self-dual threshold functions of n variables.

(b) The number of vote-assignable ND bicoteries under U is equal to the number of positive threshold functions of n variables. (Note that (P, Q) and (Q, P) are two distinct bicoteries unless $P = Q$.)

(c) The number of vote-assignable ND semicoteries under U is equal to the number of positive, self-dual threshold functions of $n + 1$ variables. □

Remark 6.1: The number of positive threshold functions of n variables $N(n)$ (which is equal to the number of positive, self-dual threshold functions of $n + 1$ variables) is bounded from below and above as follows [28, 29].

$$2^{n(n-3)/2} \leq N(n) \leq 2^{1-n} \sum_{i=0}^{n} \binom{2^n - 1}{i} (\leq 2^{n^2}).$$

Recently it has been shown in [30] that

$$\log_2 N(n) \to n^2,$$

as n tends to infinity. Comparing this with the number of positive, self-dual functions, cited in Section 2, we see that the vote-assignable ND coteries form only a very small fraction of all ND coteries. □

6.2 Multidimensional Vote-Assignable Coteries

In [8], the concept of multidimensional voting is introduced as a generalization of vote-assignable coteries. In our terminology, a family C of subsets of U is said to have an $MD(l,k)$ voting if $C = MIN(f)$ for some boolean function

$$f = F(f_1, f_2, \ldots, f_k), \tag{24}$$

where F is a threshold function with weights $w_{0i} = 1$ $(i = 1, 2, \ldots, k)$ and threshold $T_0 = l$, and each f_i $(i = 1, 2, \ldots, k)$ is a positive threshold function of $n = |U|$ variables with weights w_{ij} $(j = 1, 2, \ldots, n)$ and threshold T_i. Here, all weights and thresholds are nonnegative integers. They have shown, among other things, the following properties.

(i) Any coterie has an $MD(l,k)$ voting for some l and k.

(ii) Let $f = F(f_1, f_2, \cdots, f_k)$ be an $MD(l,k)$ voting that represents a coterie P, and let Q be the family of subsets defined by the $MD(k+1-l,k)$ voting $f' = F'(f_1^d, f_2^d, \cdots, f_k^d)$. Then, $B = (P,Q)$ is a semicoterie.

One problem here is that it is not obvious whether the family of subsets represented by an $MD(l,k)$ voting is a coterie, or whether it is an ND coterie, unless we check all subsets along the line discussed so far. An easy way to generate only coteries or only ND coteries by $MD(l,k)$ voting is given below.

Theorem 6.2:

(a) Let an $MD(l,k)$ voting (24) satisfy

- $l \geq \lceil (k+1)/2 \rceil$, and
- each f_i $(i = 1, 2, \ldots, k)$ is dual-minor.

Then $C = MIN(f)$ is a coterie.

(b) Let an $MD(l,k)$ voting (24) satisfy

- k is odd and $l = (k+1)/2$, and
- each f_i $(i = 1, 2, \ldots, k)$ is self-dual.

Then $C = MIN(f)$ is an ND coterie.

Proof. Immediate from Theorems 4.1, 2.1, and 2.2, since function F with $l \geq \lceil (k+1)/2 \rceil$ ($l = (k+1)/2$ for an odd k) is dual-minor (self-dual) as observed in Section 6.1. □

Unfortunately, it is not known whether the $MD(l,k)$ votings described in Theorem 6.2 are universal (i.e., can generate all coteries (ND coteries)) or not. However, it can be shown that $MD(l,k)$ votings satisfying the conditions of Theorem 6.2(a) are universal for coteries if and only if those satisfying the conditions of Theorem 6.2(b) are universal for ND coteries. This follows from the observation that the self-dual function f defined by (2) (Section 3) from a

dual-minor function g is a positive threshold function if and only if g is a positive threshold function.

Property (ii) above can be strengthened slightly by Theorem 2.4, since f' referred to in (ii) is equal to f^d by the fact that f is an $MD(l,k)$ voting and f' is an $MD(k+1-l,k)$ voting (see also Property (i) in Section 6.1).

Theorem 6.3 Let $B = (P,Q)$ be defined as in property (ii) above. Then B is an ND semicoterie. □

7 Conclusions

We have shown that boolean functions provide an elegant way of modeling coteries, bicoteries, and semicoteries. ND coteries, ND bicoteries, and ND semicoteries have been characterized in a very simple way in this framework, and many properties have been proved and/or reproved from these characterizations. We have also discussed decompositions of ND coteries extensively. Shannon decomposition (1) and Ashenhurst decomposition (20) provided theoretical basis for these coterie decompositions.

One open problem we are currently investigating is: to describe a given ND coterie, how should one design the corresponding network of basic majority functions? Although the lemmas in Section 4.2 give a construction of such a network, it is desirable to design an optimal network (e.g., using the minimum number of copies of the basic majority function). This is reminiscent of the minimization of the number of NAND gates in a NAND-realization of a boolean function. Furthermore, the network need not be a tree; this may help reduce the network complexity further.

Most of these design optimization problems are likely to be NP-hard. However, as in the case of logic design of computer hardware, it should be possible to develop reasonable heuristics to generate "good" networks.

Acknowledgement

We have benefited from the discussions with T. Eiter of Technical University of Vienna, M. Yamashita of Hiroshima University, M.L. Neilsen and M. Mizuno of Kansas State University, Y. Crama of Limburg University and H. Nagamochi of Kyoto University. In particular, a comment by T. Eiter has led to Theorem 4.4.

References

[1] D. Agrawal and A. El Abbadi, Efficient solution to the distributed mutual exclusion problem, *Proc. 8th PODC*, Edmonton, Aug. 1989, 193-200.

[2] R.L. Ashenhurst, The decomposition of switching functions, *Proc. Intn'l Symp. Theory of Switching*, Harvard Univ., 1957, 74-116.

[3] D. Barbara and H. Garcia-Molina, Mutual exclusion in partitioned distributed systems, *Distributed Computing 1* (1986), 119-132.

[4] P. Bertolazzi and A. Sassano, An $O(mn)$ time algorithm for regular set-covering problems, *Theoretical Computer Science 54* (1987), 237-247.

[5] L.J. Billera, On the composition and decomposition of clutters, *J. Combinatorial Theory 11* (1971), 234-245.

[6] Z.W. Birnbaum and J.D. Esary, Modules of coherent systems, *SIAM J. Applied Mathematics 13* (1965), 444-462.

[7] S.Y. Cheung, M. Ahamad, and M.H. Ammar, Optimizing vote and quorum assignments for reading and writing replicated data, *IEEE Trans. Knowledge and Data Eng.*, to appear.

[8] S.Y. Cheung, M. Ahamad, and M.H. Ammar, Multi-dimensional voting: A general method for implementing synchronization in distributed systems, Tech. Rept. GIT-ICS-89/35, School of Information & Computer Science, Georgia Institute of Technology, 1989.

[9] Y. Crama, Dualization of regular Boolean functions, *Discrete Applied Math. 16* (1987), 79-85.

[10] T. Eiter and G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, Tech. Rept. CD-TR91/16, Technische Universität Wien, 1991.

[11] A. Fu, *Enhancing Concurrency and Availability for Database Systems, Ph.D. Thesis,* Simon Fraser University, 1990.

[12] H. Garcia-Molina and D. Barbara, How to assign votes in a distributed system, *J. ACM 32*, 4 (Oct. 1985), 841-860.

[13] H. Gifford, Weighted voting for replicated data, *Proc. 7th ACM Symp. on Operating Systems*, Dec. 1979, 150-162.

[14] M.A. Harrison, *Introduction to Switching and Automata Theory*, McGraw-Hill, 1965.

[15] M. Herlihy, Dynamic quorum adjustment for partitioned data, *ACM TODS 12* (June 1987), 170-194.

[16] D. Kleitman and G. Markowsky, On Dedekind's problem: The number of isotone Boolean functions, *Trans. AMS 214* (1975), 373-390.

[17] D.E. Knuth, *The Art of Computer Programming Vol. 1: Fundamental Algorithms*, Addison-Wesley, Reading, Mass., 1973.

[18] A.D. Koršunov, Solution of Dedekind's problem on the number of monotone Boolean functions, *Soviet Math. Dokl. 18* (1977), 442-445.

[19] M. Maekawa, A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems, *ACM TOCS 3*, 2 (May 1985), 145-159.

[20] R.H. Möhring, Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and Boolean functions, *Algorithms and Software for Optimization, edited by C.L. Monma, Annals of Operations Research 4* (1985), 195-225.

[21] S. Muroga, *Threshold Logic and Its Applications*, Wiley-Interscience, 1971.

[22] M.L. Neilsen and M. Mizuno, Coterie join algorithm, *IEEE Trans. Parallel and Dist. Systems*, to appear.

[23] U.N. Peled and B. Simone, Polynomial-time algorithm for regular set-covering and threshold synthesis, *Discrete Applied Math. 12* (1985), 57-69.

[24] K.G. Ramamurthy, A new algorithm to find the smallest committee containing a given set of players, *Opsearch 25* (1988), 49-56.

[25] L.S. Shapley, On committees, in *New Methods of Thought and Procedures, edited by F. Zwycky and A.G. Wilson*, Springer, New York, 1967.

[26] D. Skeen, A quorum-based commit protocol, *Proc. 6th Berkeley Workshop on Distributed Data Management and Computer Networks*, Feb. 1982, 69-80.

[27] R.H. Thomas, A majority consensus approach to concurrency control, *ACM TODS 4*, 2 (June 1979), 180-209.

[28] R.O. Winder, Threshold logic asymptotes, *IEEE Trans. Computers C-19*, 4 (1970), 349-353.

[29] S. Yajima and T. Ibaraki, A lower bound on the number of threshold functions, *IEEE Trans. on Electronic Computers EC-14*, 6 (1965), 926-929.

[30] Yu.A. Zuev, Asymptotics of the logarithm of the number of threshold functions of the algebra of logic, *Soviet Math. Dokl. 39*, 3 (1989), 512-513.

Centre for Systems Science

• Simon Fraser University established the **Centre for Systems Science** as a provincial centre for advanced study. The objective of CSS is twofold: coordinate and combine advanced systems research at SFU, and encourage application of research results within industry in order to further the technological development of British Columbia.

• CSS has assembled three research groups to streamline its efforts. The orientation of these groups is as follows: intelligent systems, microelectronics, and computer and communications systems. Because of shared research interests, this latter group merges with LCCR (**Laboratory for Computer and Communications Research**) for the publication of technical reports.

• CSS's three groups involve over 80 skilled faculty researchers principally from SFU's Schools of Engineering Science and Computing Science, but also draw substantially on Mathematics, Physics, Geography, Philosophy, Linguistics, Psychology, Communication, and Kinesiology.